JavaScript 参考 - JavaScript | MDN (mozilla.org)

引用类型：存放在堆内存里面，值可修改

1、数组：

1. 是一类（类型）数据的集合；里面的数据称为元素，元素可以是任何类型
2. 使用中括号 [] 来快速定义一个数组；

```
var arr1 = [1, 2, 3],
    arr2 = [2, 3, 4];
console.log(arr1 == arr2);// false
```

3. 使用new Array（）来定义（创建）一个数组（正统）；

```
var 数组名 = new Array();
var arr = new Array(); //创建一个新的空数组
var age = new Array(17,18,19,20);
console.log(age);// 17 18 19 20
```

4. 数组里面的数据我们称为是元素，使用下标来访问数组的元素；
5. 数组的下标是从0开始的；

```
var fruit = ['apple', 'pear', 'banana', 'orange'];
console.log(fruit[0]);//apple
console.log(fruit[3]);//orange
```

6. Typeof 一个数组返回的是object，我们可以理解成数组是一个特殊的对象类型；

```
var arr = [1, 2, 3];
console.log(typeof arr);// object
```

7. length属性表述数组的长度；"数组名.length"访问数组的长度

```
var fruit = ['apple', 'pear', 'banana', 'orange'];
console.log(fruit.length);//4
```

2、对象：JS里面一切皆为对象

基本数据类型：string number boolean null underfined 保存在栈内存里，值不可修改，是指存放该变量的值的栈内存里面的数据不可修改；

创建一个新的对象，会在堆内存中开辟一个新的空间，变量保存的是对象的内存地址；

```
var a = 100;
    a = 200;
 console.log(a); //200
```

对象是保存到堆内存中的，元素的值可以修改；可以通过下标对数组的元素值进行修改

```
 var arr1 = [1, 2, 3];
 var arr2 = arr1; //堆内存相同
     arr2[2] = 600;
 console.log(arr1); // [1,2,600]
```

对象比较：

- 当比较两个基本数据类型的值时，就是比较值。
- 而比较两个引用数据类型时，它是比较的对象的内存地址，如果两个对象是一摸一样的，但是地址不同，它也会返回false

```
var arr1 = [1, 2, 3],
    arr2 = [2, 3, 4];//堆内存地址不同
console.log(arr1 == arr2); // false

var arr1 = [1, 2, 3],
    arr3 = arr1;//堆内存地址相同
console.log(arr1 == arr3); // true
```

多个数组使用下标获取元素：

```
  /* 二维数组*/
var arr3 = [1, [2, 3, 4], 5];
console.log(arr3[1][1]);

/*多维数组*/
var arr4 = [1, [2, [3, 4, [5, 6, ['apple', 'pear', 'banana', 'orange'], 9], 5], 1], 3];
```

```
console.log(arr4[1][1][2][2][3]);// orange
```

对象的引用，可以通过下标改变数组的元素:

```
var arr1 = [1, 2, 3];
var arr2 = arr1; //堆内存地址相同
arr2[2] = 600;
console.log(arr1); //[1,2,600]

arr1 = [100, 200, 300];//相当于创建一个新的对象，在堆内存中开辟一个新的空间，但不改变arr2的地址
console.log(arr2); //[1,2,600]
```

对象字面量:

- 使用对象字面量，可以在创建对象时，直接指定对象属性的语法：{属性名: 属性值, 属性名: 属性值...}
- 对象.属性名：属性名就是字面量；
- 对象[属性名]：中括号里面的属性名支持变量

```
var obj = { a: 1, b: 2 };
var person = {
    username: '张三',
    age: 19,
    gender: '男',
    school: '师范学院',
    major: '软件工程',
    provice: '四川',
    city: '成都',
    area: '成华区'
    };
    console.log(obj, person);

var obj1 = new Object();
console.log(obj1);
```

对象的引用：当两个变量保存的是同一个对象引用，当一个通过变量修改属性时，另一个会受到影响

```
```

```
var person = {
    username: '张三',
    age: 19,
    gender: '男',
    school: '师范学院',
    major: '软件工程',
    provice: '四川',
    city: '成都',
    area: '成华区'
};
console.log(person.username);//

var person1 = person;
person1.username = "李四";
console.log(person.username); // 李四
console.log(person1.username); // 李四
```

1. 类：是对事物的抽象
2. 对象：是类的实例化，对象属于一种复合的数据类型，在对象中可以保存多个不同数据类型的属性
3. 类和对象的关系：类是对象的抽象，对象是类的实例化；

- 类：对一群具有相同特征的对象的集合的描述；
- 对象：真实存在的对象个体
- eg：人类，指的是一个范围；对象：比如某个人，指的是这个范围中具体的对象

4. 类的特征我们称之为 属性；属性会有对应的属性值；
5. 类的动作我们称之为 方法（函数）；
6. 对象的快速定义:{属性:属性值, 属性1:属性1的值}

- 添加或修改对象属性的语法：对象.属性名=属性值；
- 读取对象属性的语法：对象.属性名；也可以用 对象["属性名"]=属性值
- 删除对象属性的语法：delete 对象.属性名；

7. 对象的正统的定义方式：new Object();
8. 对象的属性访问：

- 对象：属性名就是字面量
- 通过[ ]访问对象的属性， 支持变量 [ ]是运算符；在[]中可以直接传递一个变量，这样变量值是哪个就会读取哪个属性

```
var person = {
    'username': '张三',
    age: 19,
```

```
    gender: 'man'
  };
 //. 后面跟的属性就是字面量
console.log(person.username);
console.log(person.age);
console.log(person.gender);
//通过[]访问对象的属性  支持变量  []是运算符
console.log(person['username']);

var user1 = 'username';
console.log(person[user1]);//user1 这个变量会解析成 'username'

var g = 'age';
console.log(person['g']); //undefined  这里的g是字面量，对象没有这个属性
console.log(person[g]); //19  这里的g就是变量
```

3、什么是对象

定义：一组无序的相关属性和方法的集合，如数组，字符串，函数等

组成: 对象由属性和方法组成（可以先理解为变量和函数，后面会讲到对象属性和方法与变量和函数的区别）

js中对象该如何写：

方法一 ： 字面量创建对象

```
 var dog = {
      dogName: '可可',
      type: '阿拉斯加犬',
      age: '5岁',
      color: '棕红色',
      skill: function(){
      console.log('汪汪汪');
     }
}
console.log(dog.dogName);
console.log(dog['type']);
dog.skill();
```

方法二：利用 new Object创建对象

```
var obj = new Object(); //创建了一个空对象
```

方法三：利用构造函数创建对象

构造函数：把对象里面相同的属性和方法封装到函数里面

构造函数创建的对象可以称之为实例或者对象的实例化

（构造函数泛指的是一大类，类似java中的class）

构造函数注意点：

- 构造函数可以复用，能创建多个对象，字面量和new只能创建一个对象
- 构造函数的函数名首字母要大写
- 构造函数不需要返回return就可以返回结果，调用构造函数返回的是一个对象，但是构造函数中的方法需要返回值
- 属性和方法前面必须要加this
- 调用构造函数必须使用new

new关键字是如何实现创建对象的：

- new会先在内存中创建一个空的对象，
- 构造函数中的this都会指向这个空对象
- 执行构造函数中的代码，给空对象添加属性和方法
- 返回对象

```
/*语法*/
function 构造函数名(形参1,形参2,形参3) {
      this.属性名1 = 参数1;
      this.属性名2 = 参数2;
      this.属性名3 = 参数3;
      this.方法名 = 函数体;
}

function Movies(yourName,age,sex){
         this.name = yourName;
         this.age = age;
         this.sex = sex;
         this.skill = function(skills){
            console.log(skills);
      }
}
var cy = new Movies('杨杨',13,'男');
```

```
console.log(cy.age);
console.log(cy['sex']);
cy.skill('hulalala');
```

遍历对象

```
var dog = {
    dogName: '可可',
    type: '阿拉斯加犬',
    age: '5岁',
    color: '棕红色',
    skill: function(){
    console.log('汪汪汪');
    }
}
//for (变量 in 对象) 变量一般使用k或者key
for (var key in dog){
console.log(key); //key变量输出得到的是属性名和方法名
console.log(dog[key]); //dog[key] 得到的是属性值;
//key是个变量，要想拿到对应的属性值，需要使用[]。因为[]是支持变量的，不能使用.拿到属性值
}
```

变量和对象的属性

- 相同点：都是用来保存数据的
- 区别:

  变量: 单独声明并赋值，使用的时候直接写变量名，单独存在

  对象的属性: 在对象里面不需要声明，使用的时候必须是对象.属性

函数和对象的方法

- 相同点:都是实现某种功能做某件事情
- 区别:

  函数: 单独声明，并且调用是单独存在的

  对象的方法: 调用的时候，对象.方法( )